Meteorology Data Nodes

A cheap solution using Raspberry Pis

Jt Whissel

2012

The process of gathering weather data by the use of technology has been around for decades. However with the ever expanding growth of its advances, the cost of the equipment for collecting the data has not diminished. This lack of decrease in equipment costs has no justification behind it when available resources, though not complete, could be fabricated together to create the same or even better system.

Contents

Introduction 2

Existing Products 3

Davis Instruments 6152 Wireless Weather Station 3

Netatmo 3

Ambient Weather WS-1090 4

Oregon Scientific WMR200A 4

WeatherHawk 916 Wireless Weather Station 5

MK-III-LR 5

Proposed Design 6

Functional Specification 6

Scope Objectives 6

Comparison of Raspberry Pi to Arduino 7

ATMega328P 10

Raspberry PI 10

ShiftBrite 11

BMP085 11

TEMT6000 11

HH10D 11

HMC6352 11

DS18S20 12

DS2745U+ 12

Measuring UV 12

Current Work Progress 13

Raspberry PI 13

ATmega328P 14

Works Cited 15

Introduction

A weather station is used to collect different metrological data to help forecast the weather. Since technology advances so fast, it has enabled personal weather stations to be bought and used in homes, schools and other personal places. Some of these weather stations can be advance packed with copious amount of sensors that the general user would not need. However, there are personal weather stations that come with just the right amount of sensors to enable the user to have a general idea what the weather is going to be. The only problem with a lot of these personal weather stations is the price for the amount of functionality it brings. This is the niche we are going to try to fill with our personal weather nodes.

The weather nodes will be little translucent white cubes that can be dispersed around to collect accurate real-time weather data. The weather nodes all interconnect wirelessly to each other creating a network of nodes to collaborate as one unit. This allows the end user to collect different kinds of weather data per node. Each node will have the ability to hook up one of the custom sensors that are specifically made for the weather nodes. However, the user will also have the ability to use other sensors that work with Linux, by interfacing with the weather node software.

Each one of these nodes will be able to be configured to the user's needs via an in-browser configuration page from the designated master node. Users could use only a single node and still have a full, functioning system. Although you can connect as many slave nodes as you want to a single master node. Each node has the ability to be toggled to become a master or a slave node, making the possibilities for multiple master nodes to create isolated data collection.

The nodes feature an interesting way to relay information to the user by using colors. Since the whole node is a white translucent cube, it will use a RBG led to defuse color onto the cube. An example of how a user would get information from the cube via color would be if the user was setting up a salve node from a master node and did not know the range it could communicate in. The cube would be green for excellent signal and accurately transitioning into red for being completely out of communication rage.

Existing Products

Davis Instruments 6152 Wireless Weather Station

The Davis Instruments 6152 has a large amount of sensors for tons of functionality packed into its system. It is broken up into two devices, one being the wireless console and the other is the wireless sensors that can be placed all together from up to 300 meters in line of sight. The system comes with sensors for humidity, rain, temperature, wind, direction and barometric pressure for a starting price of $535.95. [1] Additionally you can get optional sensors that will allow for measuring evapotranspiration, leaf wetness, soil moisture, solar radiation, UV radiation and UV meds.

The wireless console has a 6 x 3 ½ inch LCD grayscale display to view the weather data on. The screen has on screen graphing for averages, current conditions or highs and lows for the last 24 hours, days, months or years. It also contains all the information from the sensors displayed with a 2 ½ second delay. The console also has the ability to be connected to a pc to view and log the weather data via serial interface (WeatherLink). Software from Davis for communication to the console can be purchased for additional $165 [2]. However, there is some cheaper software from a 3rd party developer that has all the features of WeatherLink and additional ones like sending custom weather updates to your twitter status. [3]

Netatmo

Netatmo claims to be “The first Made for iPhone Weather Station” [4] making it have a interesting market place for personal weather stations. For $179.00 you get the IOS app and two wireless modules. The indoor module measures temperature with +-0.3°C accuracy, humidity with +-%3 accuracy and C02 with +-5% accuracy. The outdoor module measures temperature with +-0.3°C accuracy, humidity with +-%3 accuracy , barometric pressure with +-1mbar and sound dB with the range of 35db to 120db. The whole system gets updated every 5 minutes to give you updated data.

Ambient Weather WS-1090

The WS-1090 is a compact, feature rich and cheap personal weather station going only for $99.99. [5] The system has sensors to measure wind speed, wind direction, temperature, humidity, rainfall and barometric pressure. The data from these sensors is updated every 48 seconds wirelessly to the 5.75 x 4.25 inch grayscale touchscreen weather console. The weather console also has a usb port to connect to your pc to view the data with their free software. This will allow you to save as many sample intervals as you want if you record them on your pc. If you wanted to record them on the system itself, it is able to handle up to 4080 data points.

Oregon Scientific WMR200A

The WMR200A is a midrange personal weather station that has some interesting features. It comes with a standard wireless touchscreen grayscale console that displays its data with. The WMR200A also comes standard with sensors to measure humidity, temperature, wind chill, wind speed, wind direction, barometric pressure and rainfall. However for the price of $349.99 [6] the sensors are surprising not accurate and the base only gets a update every 60 seconds. For example the temperature accuracy is +- 2°C and the humidity sensors are up to 7% off [7]. Nonetheless the whole system can also connect up to 10 wireless sensors up to 300 feet away. To power those wireless sensors it also comes with a solar panel to power them along with their battery backup. A nice feature that this system has the ability to read WWVB-60 signals to get the time from the atomic clock at Fort Collins, Colorado.

Even though its sensors may not be that accurate, the system makes up for it in its software. It has the ability to be hooked up to the PC via a usb cable. This will make the system communicate with the free software that it comes with called Virtual Weather Station (Base Edition). This software displays any data from the console on your computer. It allows you to record and save data as well as stream it to weather underground where you can view you weather from the web.

WeatherHawk 916 Wireless Weather Station

The WeatherHawk 916 is considered a professional personal weather station. It runs for a starting price of $2,745 [8], but it has features that other personal weather stations do not. It comes with sensors to measure air temperature, relative humidity, barometric pressure, rain, solar radiation, wind speed and wind direction. The sensors accuracy is quite impressive with temperature +- 0.5°C, humidity +- 3% and rain resolution at 1mm. The system does not come with a base station with a console display; instead it comes with a wireless receiver base that plugs into your computer. The weather station has a range of ½ mile while within line of sight and over 7 miles with optional external antenna configurations. The WeatherHawk 916 also has the ability to store around 32,000 data points inside of its 128 kbytes of nonvolatile Flash RAM. This is the only weather station in this paper that claims to have real-time weather data. This is contrary to other weather stations that can have over a minute of wait time before weather data gets updated.

MK-III-LR

The MK-III-LR is a long range wireless weather station costing around $995 [9]. It provides data from up to 1 mile away within line of sight. It also has a close to real-time update interval of 2 seconds. The system comes with an array of accurate sensors including wind speed, wind direction, temperature, relative humidity, barometric pressure and rain fall. Compared to other systems its sensors have high accuracy, for example the temperature sensor is +- 0.25°C and the humidity sensors accuracy is +- 2%. The station also comes with a solar panel and battery that they guarantee 60 days of operation without out sun. Just like the WeatherHawk 916 this system comes with a wireless receiver that you can hook up to your pc to view the data.

Proposed Design

Functional Specification

Scope Objectives

Hardware

- ATMega328-PU
- Raspberry PI
- ShiftBrite (SPI)
- DS2745U+
    - I2C address = 0x48
- Barometric Pressure Sensor - BMP085 (I2c)
    - I2C address = 0x77
- Ambient Light Sensor - TEMT6000 (Analog)
- Photodiode
- Humidity sensor - HH10D (I2C)
    - I2C address = 0x51
- Compass Module - HMC6352 (I2C)
    - I2C address = 0x42
- Temperature sensor – DS18S20 (1-Wire)
- Custom Anemometer (Analog)
    - Using either a small brushless motor
    - Or using slotted infra-red sensor
- White Translucent Cube to house the weathers bread board and raspberry pi
    - An unknown sized LIPO with solar panel on the top of the cube.

Software

- Custom Debian Linux distribution
    - Minified for maximum ram
    - Pre install packages to make everything work together
    - Node.js as the server for the whole cube
    - Auto configures its Wi-Fi based on what the user wants.
- GUI
    - Will be able to be accessed in browser by connecting to its WIFI.
    - Shows welcome screen when user firsts connects and asks questions to auto configure all the nodes.
    - Auto detects the nodes and shows each one and its sensors data real time.
    - When a new weather node connects it will create a node in the GUI in real time.
    - No refreshing is required since the webpage is dynamic.
    - Ability to show numeric data or graph data for each node.
    - Able to click on a node to get a full screen view of the graph

    - Each node has a record button to record its data in real time.
    - For each recorded session you are able to export the data in CVS or XML formatting.
- Server
    - Auto detects what sensor is connected and then has it configure the data correctly to transmit the data to the master server.
    - Knows if it's the master node or slave node via toggle switch.
    - If it's the master node it configures its self to be an AP so slave nodes can connect to it its wifi and server.
    - The master node also serves the webpage to the client via web browser.
    - Master node will also contain all the saved data sessions that the user wishes to save.

Comparison of Raspberry Pi to Arduino

The table below shows the comparison of the two systems that we researched to possibly use as our solution while making the weather nodes. This data shows the Raspberry Pi has extreme power advantages to the Arduino. However, the Arduino has more Digital I/O pins that support PWM and has analog inputs.

Features	Raspberry Pi	Arduino Uno
Processor	ARM1176JZF-S (armv6k)	ATMega328P
Clock Speed	700 MHz	16 MHz (8MHz without crystal)
RAM	256Mb (SDRAM)	2Kb (SRAM)
Digital I/O Pins	17 (of which 1 provides PWM output)	14 (of which 6 provide PWM output)
Analog Input Pins	0	6
Power Consumption	700mA	12mA
Price	$35	$30

Arduino Uno

For this comparison we used the most current project board by Arduino called "Arduino Uno". The Arduino Uno provides a platform for programming embedded applications to the ATmega328. Running with the clock speed of 16 MHz the ATmega328 also provides eight GPIO[1]s and six PWM[2] outputs, which also dual function as six more GPIOs. The board also includes six analog inputs, an ICSP[3] header and a micro USB connection to communicate via serial to device.

[1] GPIO – General Purpose Input Output
[2] PWM – Pulse Wave Modulation
[3] ICSP – In-Circuit Serial Programming

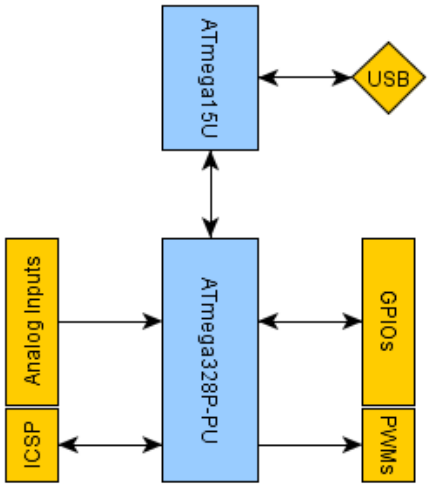

Figure 1 – Block Diagram of the Arduino Uno [10]

The way users would communicate with the ATmega328 chip would be through the ATmega15U that is taking care of the USB to serial communication between the two chips. Arduino has developed an IDE for easy programing the ATmega328 via this communication. [11]

Analog Inputs

The pins (23-28) on the Atmega328 are dedicated to Analog input. For example one can use a potentiometer that could be hooked up to 5v power, ground and one of the analog inputs to get a reading, noting that the max voltage for the analog inputs is 5v. The way the reading works is through a built in analog-to-digital converter, the voltage first comes in between 0-5v. Then it is converted to digital, in the form of a number between 0-1023. This number range is the same range when dealing with the Atmega328 PWNs.

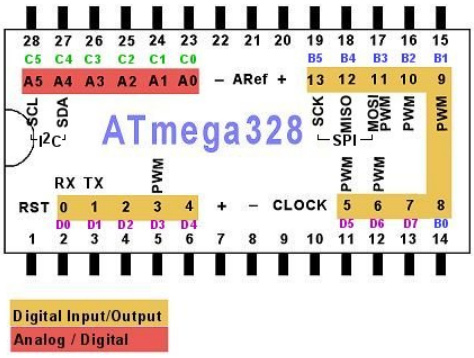

Figure 2 - ATmega328's pin out the diagram to visualize the inputs and outputs. [12]

ATmega328 CPU

The Atmega328 is based on the Harvard architecture 8-bit RISC. This processor has an internal clock speed of 8Mhz. However, when connected to an external (16Mhz – 22Mhz) crystal you are able to get those speeds by flashing an edited version of the Arduino boot loader. The default for the Atmega328 is to use the external clock. The chip uses the Atmel AVR instruction set with each instruction being 16 bits long.

Raspberry Pi

The Raspberry PI is a full computer that is the size of a credit-card. This tiny computer two USB ports, Ethernet, HDMI, RCA Video, 3.5 mm Audio jack, SD Card slot and an array of GPIOs. The computer has 256MB of ram that is shared between the CPU and GPU. This gives the computer the ability to display 1920x1080 through the HDMI without any performance issues. The most compelling feature of this computer is that its price is set for only $35.

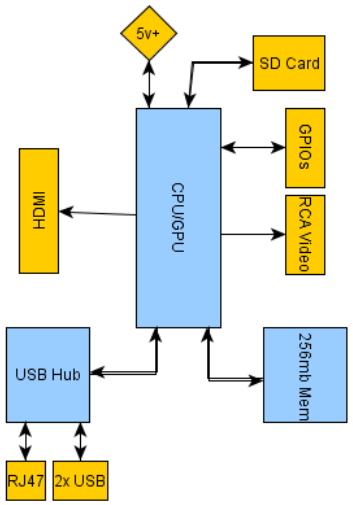

Figure 3 - Block Diagram of the Raspberry Pi [13]

Raspberry PI CPU/GPU is an armv6k clocked at 700Mhz, this gives an extreme power lift compared to the speed of a ATMega328. However, it does lack the interface pins that the ATMega328 has. The Raspberry PI only has one functional PWN (without reusing some of the PWMs that are used for audio), this makes interfacing with I2C, SPI and other hardware interfaces hard to do unless you wanted to bit bang the protocols.

This small computer can use any operating system that supports ARM architecture. The armv6k also support operating systems that are complied with ARM hard float. This will give the whole system better performance since this CPU utilizes hard float. Without hard float, the operating system tries to software emulate floating point hardware.

ATMega328P

While the ATMega328P is not the centralized component to the node, it is the one that actually communicated with all the sensors to be able to relay the information to the raspberry pi. The basics for how the ATMega328P will serve the information to the raspberry pi is inside its code, it will constantly be listening on two of the analog inputs to sense if there is any voltage change. When the voltage is zero then there is no device hooked up to the node, this will let the ATMega328P know that it now needs to listen for a device to connect. When a device connects it will put power to what one of the analog pins with a unique voltage to the sensor. This is how we can tell what device is connected to run the proper code to update the Raspberry PI with new information. After the device connects, it then reads the information from the sensor as fast as the sensor will allow it to be read. If there are any changes to the data received from the sensor, it then pushes out the new data through serial to the raspberry pi. This is done with all of the listed sensors expect for the analog sensors in this paper using I2C or 1-Wire protocol.

The ATMega328P controls the analog sensors similar way it senses what device is connected if it is connected regarding to the digital sensors. The chip listens to see if there is a change in voltage to know when the sensor is plugged in. Then it starts read the sensor every update tick to provide real time information back to the Raspberry Pi when there is a change in data. The only difference between how these sensors are detected for a connection from the digital sensors is that the analog sensors do not need to supply voltage through a resistor to provide a way of identifying what it is. This is because there will be a separate plug for the analog sensor.

Raspberry PI

The Raspberry PI will be the central part of this project. It will be used to host the web client, communicate between the ATMega328P and to handle all of the configuration and data storage. The design flow of the communication between the Raspberry PI and the ATMega328P is through serial communication. This utilizes pins 14 (TXD) and 15 (RXD) on the Raspberry PI. Since the Raspberry PI does not contain any over-voltage protection and is not 5v tolerant we had to put 1k Ohm resistor between pin 15 (RXD) and the ATMega328P's TX line. We also had to put another 1k Ohm resistor between pin 15 and ground making the voltage not go over 3.3 volts.

On the networking side the Raspberry PI uses two usb wireless cards that support G,B and N. The chip that runs on each of the wireless cards is a Realtek 8192CU. This particular chip was not supported by the distribution of Linux that is being used for this project (Raspbian Wheezy compiled for ARM with Hard Float binaries). This created the challenge of getting that chip to work on that distribution without switching to a more expensive wireless card. However, we got the wireless cards to work by finding the kernel files for it and configuring it into the system manually. The way the project utilizes these wireless cards is having one connecting to access point while the other creating its own access point and then bridging them together. After the user configures the master node, it will be able to scan and connect to other local access points around the node to connect with to enable internet

access through the web client interface.

The last role the Raspberry PI has is to host and run the client's web interface. This is done with a series of programs written in java script, html5 and C. The server's backend is the central part of the web client design. One of the responsibilities for the server is to serve the client to people accessing the node within a web browser. Another responsibility is to handle all of the communication between the ATmega328P to listen to the data and update the client when needed. The server also handles all of the saving and loading of the clients data, while also controlling the ShiftBrite through the ATmega328P.

ShiftBrite

ShiftBrite is a module containing a 140 degree viewing angle RGB LED with 8000mcd per color. It also has on the module the A6281 which is the IC that controls the RGB with 10-bit PWM for each color [14]. The module uses SPI for communication to the A6281 to control the RGB, making it a perfect solution to use with the Raspberry Pi since it lacks PWM pins to control a RGB LED.

BMP085

The Bosch BMP085 is a low-power and high-precision barometric pressure sensor. Its accuracy can be +- 0.03 hPa while measuring a range of 300 to 1100 [15]. This sensor also has a built in temperature sensor included inside the IC. The protocol to talk to the BMP085 is I2C making it only need two wires to communicate with the IC.

TEMT6000

This is a popular light sensor that you can find in a lot of consumer electronics today including your cellphone or notebook. It has a spectral bandwidth range of 360nm to 970nm with a wavelength of peak sensitivity of 570nm [16]. This sensor uses analog output to relay its data to a microcontroller.

HH10D

The HH10D is a Humidity sensor that has high accuracy of +-3% relative humidity. This sensor uses I2C protocol to communicate its data through. This particular module has a capacitive type humidity sensor, a CMOS capacitor to frequency converter and a EEPROM used to hold calibration factors [17]. The humidity sensor also has an impressive resolution of 0.3% relative humidity.

HMC6352

The Howneywell HMC6352 is a compass module that uses 2-axis magneto-resistive sensors, algorithms for heading computation and its onboard firmware to give back the heading through I2C protocol [18]. This compass module has a heading resolution of 0.5 degrees. This module also features stray magnetic field protection and temperature compensation.

DS18S20

This high-precision digital thermometer by maxim integrated uses a unique protocol called 1-Wire. This interface only requires 1 wire to for sending and receiving data through. On top of that, that same data line can be uses to power this IC through a process they call parasitic power making it a true 1 line device. The thermometer has high accuracy of +- 0.5°C with 9 bits of resolution [19]. Just like most 1-wire devices from maxim integrated, this sensor has a unique 64 bit unique serial code to address it by. This makes it possible to run multiple 1-wire devices with still only one wire.

DS2745U+

The DS2745U+ is an 8 pin IC that can measure voltage, current-flow and temperature. This IC id designed for monitoring batteries in one low cost single chip. This chip comes in a SOIC form, making it not able to be dropped into a bread board for testing like a DIP chip would allow. We had to make a break out board for this small chip that is about the size of a grain of rice placed diagonally along the chip. After creating the board we used I2C protocol to communicate with the chip with the ATMega328P.

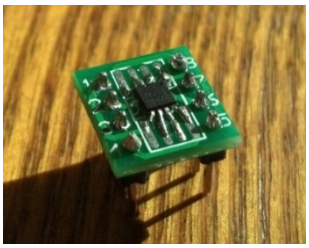

Figure 4 – A photo of the board we created to break out the pins to enable communication.

To start communication with the chip you first need to transmit its hardware address, which by default is 0x48 in hex. This how ever has a programmable address that can be changed after the first communication is established. After the request for transmitting has been connected, then we wrote the memory address that we wanted to read from. On this chip, there is two address for each measurement. For example if we wanted to get the temperature, we would have to read from addresses 0x0A (MSB) and 0x0B (LSB).

Measuring UV

Every weather station seems to give UV index reading. We have the TEMT6000 for light sensing, but that sensor, even though it goes as low as 360nm in the spectral range, its lens protects it from UV and IR waves. [16]

To actually calculate the UV index it requires extremely expensive sensors ranging from $50 to $1000+, and the use of two satellites operated by the National Oceanic and Atmospheric Administration measuring the current total ozone amounts over the whole globe. [20] The full spectrum range to measure is 290nm to 400nm, which is the full spectrum of UV-A and UV-B. However, this leaves out other radiation the sun gives off which should be a part of the UV index calculation, such as UV-C. This could be because of the very low amount that reaches the earth and the fact that UV-C's wavelength range is extremely low (280nm-100nm).

Noticing the fact the sensors just for UV index sensing cost $50 -$ 1000, it defeats the purpose of this project which is to make a cheap solution to meteorology data collection. Nonetheless, we discovered the ability to use LEDs as photodiodes. This could make it possible to buy cheap UV LEDs and use them as photodiodes. The way this works is by hooking up the LED's negative lead to positive

voltage and the positive lead to low voltage. This will charge up the capacitance of the LED making the LED not come on. After that, we will cut off the voltage of the negative lead and count how long it takes for the voltage to drop to logic zero giving us how intense the light is.

Current Work Progress

Raspberry PI

We researched the Raspberry PI to better understand how it works to be able to get it to work properly. The first thing we did was research what operating system was most fitting for our needs while keeping in mind the limitations of the device and the scope of our project. After trying a few of them we decided to go with the Debian Wheezy armel distribution complied with hard float binaries and striped of every package that is not necessary to run [21]. This makes the image only 109MB and only uses 7MB of ram after boot.

After we got the OS flashed onto the SD card we noticed that the memory card would randomly get corrupted making it not possible to boot anymore unless you re-flashed the OS. We thought this could be because of poorly shutting down the device (by ripping the power cord out), or possibly having the SD card get static shocked since the SD card's pins are exposed on the bottom of the device. However, this proved to be a problem enough that we wrote bash scripts to back up the current image and then a script to flash that image back to the SD card.

Our first major task that was completed on the Raspberry PI was to get the wifi drivers to work with the system. Since the drivers were not in the stock kernel, we had to compile the drivers for arm and then load the drivers manually. After that we still needed to configure the wireless to work with both cards that we have put on the device. We configured one of the cards to work as a infrastructure to connect to the internet while the other one acts like a AP host for the other cubes and user to connect to.

Interfacing with IC components was a new subject for us, making our task of getting a RBG led to work using the GPIOs from the Raspberry PI a vigorous task. With tons of trial and error we then learned all about PWM and how the Raspberry PI only has one accessible PWM output. This discovery lead to a module called the Shiftbrite that controls the RGB led from an IC that communicates through SPI protocol. After copious amount of hours of trying to "bit bang" the SPI protocol with C code, we finally got the Shiftbrite to work from the Raspberry PI.

ATmega328P

After we gained some knowledge of how the basics of talking to hardware works, we moved on to learning about the ATmega328P. We learned about coding on the chip itself and started to make simple programs to read in analog input and utilizing the digital output pins to turn on and off LEDS. Then we wanted to start actually talking to some hardware. This was not as hard as it was on the Raspberry Pi since the ATmega328P supports a lot of protocols that ICs use like SPI, I2C and serial. The first chip we got working was the DS18S20 temperature sensor using 1-wire protocol. The Arduino IDE has a library for communicating to 1-wire devices. However the library only helped with the physical communication, we still needed to read the datasheet and figure out about the Scratchpad and what memory locations to call. At first we could not even read the address from the device indicating that either we fried the chip or our wiring was wrong. After searching more through the data sheet we discovered that we needed a 4.7k pull-up resistor on the data line to 5v rail. Adding the pull-up resistor made the communication work.

When we successfully got communication to work on the DS18S20, we then began getting communication to work via serial between the raspberry pi and the ATmega328P. The Raspberry PI uses 3.3v serial communication while the ATmega328P uses 5v serial. However, the ATmega328P can listen just fine without the need of a RS-232 conversion. The only thing we added was two 1 kOhm resistors between the Raspberry PI's RX and the Atmega328P's TX to form a voltage divider. This made the connection work almost perfectly after we tried different bud rates. We found that 9600 bud rate worked the best for the Raspberry PI since that is what its startup default bud rate is set at.

Works Cited

[1] Weather Shack, "Davis Vantage Pro2 Weather Station," [Online]. Available: http://www.weathershack.com/davis-instruments/davis-vantage-pro2.html?gclid=CPXDpNPUs7MCFRRynAodZ2sAbQ. [Accessed 3 11 2012].

[2] Davis, "WeatherLink, Windows, USB," [Online]. Available: http://www.davisnet.com/weather/products/weather_product.asp?pnum=06510USB. [Accessed 12 11 2012].

[3] Afterten Software, "WeatherTracker," [Online]. Available: http://www.afterten.com/products/weathertracker/. [Accessed 12 11 2012].

[4] Netatmo, "Netatmo," [Online]. Available: http://www.netatmo.com/en-US/product. [Accessed 13 11 2012].

[5] Ambient Weather, "Ambient Weather WS-1090 Wireless Home Weather Station," [Online]. Available: http://www.ambientweather.com/amws1090.html. [Accessed 12 11 2012].

[6] Ambient Weather, "Oregon Scientific WMR200A Wireless Solar Powered Professional Weather Station," [Online]. Available: http://www.ambientweather.com/orwmr200.html. [Accessed 25 11 2012].

[7] Oregon Scientific, "WMR200A," [Online]. Available: http://www2.oregonscientific.com/ulimages/manuals2/WMR200A.pdf. [Accessed 25 11 2012].

[8] WeatherHawk, "WeatherHawk 916 Wireless Weather Station," [Online]. Available: http://www.weatherhawk.com/s916w. [Accessed 26 11 2012].

[9] RainWise Inc., "MK-III RTI-LR," [Online]. Available: http://www.rainwise.com/products/detail.php?ID=6801#prod-support. [Accessed 26 11 2012].

[10] Arduino, "Arduino," [Online]. Available: http://arduino.cc/en/Main/ArduinoBoardUno. [Accessed 13 9 2012].

[11] Arduino, "Download the Arduino Software," [Online]. Available: http://arduino.cc/en/Main/Software. [Accessed 8 11 2012].

[12] hobbytronics, "Arduino ATmega328 Pinout," [Online]. Available: http://www.hobbytronics.co.uk/arduino-atmega328-pinout. [Accessed 9 11 2012].

[13] Wikipedia, "Raspberry Pi," [Online]. Available: http://en.wikipedia.org/wiki/Raspberry_Pi. [Accessed 23 9 2012].

[14] Macetech, "Shiftbrite," [Online]. Available: http://docs.macetech.com/doku.php/shiftbrite. [Accessed 28 11 2012].

[15] Sparkfun, "DMP085," [Online]. Available: https://www.sparkfun.com/products/11282. [Accessed 29 11 2012].

[16] Vishay, "TEMT6000," [Online]. Available: http://www.sparkfun.com/datasheets/Sensors/Imaging/TEMT6000.pdf. [Accessed 31 10 2012].

[17] Hope RF, "Humidty Sensor Module," [Online]. Available: http://www.hoperf.com/upload/sensor/HH10D.pdf. [Accessed 29 11 2012].

[18] Honeywell, "2-Axis Compass with Algorithms HMC6352," [Online]. Available: http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Missiles-Munitions/HMC6352.pdf. [Accessed 29 11 2012].

[19] Maxim Integrated, "DS18S20 High-Precision 1-Wire Digital Thermometer," [Online]. Available: http://datasheets.maximintegrated.com/en/ds/DS18S20.pdf. [Accessed 29 11 2012].

[20] United States Environmental Protection Agency, "How UV Index is Calculated," [Online]. Available: http://www.epa.gov/sunwise/uvicalc.html. [Accessed 2 11 2012].

[21] LinuxSystems, "Raspbian Wheezy armhf Raspberry Pi minimal image," [Online]. Available: http://www.linuxsystems.it/2012/06/raspbian-wheezy-armhf-raspberry-pi-minimal-image/. [Accessed 10 6 2012].

[22] Atmel, "8-bit AVR Instuction Set," 2012. [Online]. Available: http://www.atmel.com/images/doc0856.pdf. [Accessed 29 9 2012].